

SVM

Svm算法：

1.定义超平面，以它分类：

$$h(x) = w^T x + b$$

分类方法：

$$\begin{cases} w^T x_i + b \geq 1 & y_i = +1 \\ w^T x_i + b \leq -1 & y_i = -1 \end{cases}$$

2.间隔最大化：

$$\rho = 2r^* = \frac{2}{\|w\|}$$

间隔

最大化 ρ 值需要有条件限制，即：

$$\begin{cases} \max_{w,b} \frac{2}{\|w\|} \\ y_i(w^T x_i + b) \geq 1, (i = 1, \dots, n) \end{cases}$$

用拉格朗日乘数法求解：

$$L(x) = f(x) + \sum \alpha g(x)$$

3.我们的目标：

$$\min_{w,b} \max_{\alpha \geq 0} L(w, b, \alpha)$$

推导结果：

$$W(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j x_i^T x_j$$

$$\sum_{i=1}^n \alpha_i y_i = 0, \alpha_i \geq 0 (i = 1, \dots, n)$$

Ktt条件：

$$\begin{aligned}\alpha_i = 0 &\Rightarrow y^{(i)}(w^T x^{(i)} + b) \geq 1 \\ \alpha_i = C &\Rightarrow y^{(i)}(w^T x^{(i)} + b) \leq 1 \\ 0 < \alpha_i < C &\Rightarrow y^{(i)}(w^T x^{(i)} + b) = 1.\end{aligned}$$

优化 α^* ，最后得出

$$w^* = \sum_{i=1}^n \alpha_i^* y_i x_i, \quad b^* = 1 - w^{*T} x_s$$

4. 优化 α^* 的具体算法：SMO算法

每次循环中选择两个 α 进行优化处理。一旦找到一对合适的 α ，那么就增大其中一个同时减小另一个。这里所谓的“合适”就是指两个 α 必须要符合一定的条件，条件之一就是这两个 α 必须要在间隔边界之外，而其第二个条件则是这两个 α 还没有进行过区间化处理或者不在边界上。

α_1, α_2 用其他参数表示

$$\alpha_1 y_1 + \alpha_2 y_2 = - \sum_{i=3}^n \alpha_i y_i = \zeta$$

α_1 用 α_2 表示

$$\alpha_1 = (\zeta - \alpha_2 y_2) y_1$$

带入该式再求解 α_2 得 α_2^*

$$W(\alpha) = W(\alpha_1, \alpha_2, \dots, \alpha_n) = W((\zeta - \alpha_2 y_2) y_1, \alpha_2, \dots, \alpha_n)$$

而这个 α_2^* 需要满足一个条件 $L \leq \alpha_2^* \leq H$

当 y_1 和 y_2 异号时：

$$\begin{cases} L = \max(0, \alpha_2 - \alpha_1) \\ H = \min(C, C + \alpha_2 - \alpha_1) \end{cases}$$

当 y_1 和 y_2 同号时：

$$\begin{cases} L = \max(0, \alpha_2 + \alpha_1 - C) \\ H = \min(C, \alpha_2 + \alpha_1) \end{cases}$$

最终值 α_2^{**} ：

$$\alpha_2^{**} = \begin{cases} H & , \alpha_2^* > H \\ \alpha_2^* & , L \leq \alpha_2^* \leq H \\ L & , \alpha_2^* < L \end{cases}$$

再通过 α_2^{**} 更新 α_1

5.迭代过程：

重复下面过程直到收敛{

(1) 选择两个拉格朗日乘子 α_i 和 α_j ；

(2) 固定其他拉格朗日乘子 $\alpha_k(k \neq i \text{ 和 } j)$ ，只对 α_i 和 α_j 优化 $w(\alpha)$;

(3) 根据优化后的 α_i 和 α_j ，更新截距 b 的值；

}

每次迭代选取两个 α_i

那选择的第一个变量 α_i 就选违法KKT条件最严重的那一个。

在选择第一个 α_i 后，算法会通过一个内循环来选择第二个 α_j 值。因为第二个乘子的迭代步长大致正比于 $|E_i - E_j|$ ，选 $|E_i - E_j|$ 最大的 α_j 。

6.这样就可以两分类了。

对于mnist 10分类的算法，我的做法是使用10个svm，每一个分出是某个数值和不是某个数值，从0-9的顺序依次比较，直到判断是。

这样比较的次数少但是对于向后的数字不公平，会产生误差。